

WES-CS GROUP MEETING #8

Exercise 1: You're Driving me Crazy! (Simulating Java Code)

This exercise will help you to understand what happens when objects are declared and created, and when methods are called.

It will also help you to understand the difference between copying from one variable to another when the variable is an object, and when it is a primitive type (int, double, boolean, etc), as well as the difference between changing the values of variables that are objects vs primitive types.

First, take a look at the Car class defined on the next page.

Now execute the code fragment below; let one person play the role of each variable (myCar, yourCar, anotherCar, oldSpeed, and currSpeed), and also have one person be the Java memory manager. When a variable is assigned to, or one of its methods is called, the person playing the role of that variable should act out the effects of the assignment or call. The memory manager should handle calls to new and should also reclaim any memory that gets freed.

```
Car myCar, yourCar, anotherCar;
int oldSpeed, currSpeed;

myCar = new Car("beep");
yourCar = new Car("honk");
anotherCar = myCar;

currSpeed = myCar.getCurrSpeed();
yourCar.changeSpeed(7);
anotherCar.changeSpeed(20);
currSpeed = myCar.getCurrSpeed();

myCar.blowHorn(2);
yourCar.blowHorn(3);
anotherCar.blowHorn(4);

oldSpeed = currSpeed;
myCar = yourCar;
currSpeed = myCar.getCurrSpeed();

myCar.changeSound("ooga");
myCar.blowHorn(currSpeed/5);
yourCar.blowHorn( yourCar.getCurrSpeed()/2 );
anotherCar.blowHorn( myCar.getCurrSpeed()/10 );
anotherCar = myCar;
```

```

public class Car {
    /*******
     * data members
     *****/
    private int currSpeed;
    private String hornSound;

    /*******
     * public methods
     *****/
    /* constructor */
    public Car(String sound) {
        currSpeed = 0;
        hornSound = sound;
    }

    /* changeSound: change the horn sound */
    public void changeSound(String newSound) {
        hornSound = newSound;
    }

    /* blowHorn: blow the horn;
     * parameter numTimes tells you how many times
     */
    public void blowHorn(int numTimes) {
        while (numTimes > 0) {
            System.out.println(hornSound);
            numTimes--;
        }
    }

    /* changeSpeed: change speed */
    public void changeSpeed(int milesPerHour) {
        currSpeed = currSpeed + milesPerHour;
    }

    /* getCurrSpeed: return the current speed */
    public int getCurrSpeed() {
        return currSpeed;
    }
}

```


Exercise 3: PigLatin

The rules for translating an English word to Pig-Latin are as follows:

- If the word starts with a vowel, it is unchanged.
- Otherwise, all of the consonants at the beginning of the word (up to the first vowel) are moved to the end of the word, preceded by a dash (for readability), and followed by “ay”.

For example, the sentence “I love Madison in the springtime” would be translated to “I ove-lay adison-May in e-thay ingtime-spray”.

For this exercise, you will complete the *PigLatin* class started on the next page, which will allow users of the class to translate an English word to Pig-Latin.

First, make sure you understand the translation rules by translating the following phrases into Pig-Latin:

Hello world

I love WESCS

the car says ooga ooga

Part (a): Take a look at the incomplete *PigLatin* class on the next page. Make sure you understand what each method is supposed to do, and discuss the reasons for making some of the methods private and others public.

```

public class PigLatin {
    private String englishWord;

    // constructor
    public PigLatin( String s ) {
        englishWord = s.toLowerCase();
    }

    // translate: translate the English word to Pig-Latin
    public String translate( ) {
        /* part (c) */
    }

    // firstVowelPos: return the position of the first vowel
    //                  in the English word; return -1 if there
    //                  is NO vowel in the word
    //
    private int firstVowelPos( ) {
        /* part (b) */
    }
}

```

Part (b): Divide into groups, each working on one laptop. Write the *firstVowelPos* method, which returns the position of the first vowel in the *englishWord* field, or returns -1 if there is no vowel in the word. Here are some examples:

englishWord	Result of calling <i>firstVowelPos</i>
hello	1
ice	0
spring	3
qyzzx	-1

To test your code, write a *main* method for the *pigLatin* class that creates several *PigLatin* objects, calls their *firstVowelPos* methods and prints the results.

Part (c): Now describe in English how the *translate* method of the *PigLatin* class should work (using the rules for translating from English to Pig-Latin given above, and making use of the *firstVowelPos* method). Then write the actual code. Also write a *TestPigLatin* class with a *main* method that asks the user of the program to type in a word, reads the word, and prints the translation. Compile your code and run it!

Exercise 4: Time is of the Essence! (Logical Thinking)

1. You've been lost in the African savanna for days and you're starving. As luck would have it, you stumble upon a shack containing a fresh ostrich egg and a warning: Boil this egg for exactly 9 minutes!

Fortunately there is a pot, water, salt, and a wood stove with supplies to start a fire, but you don't have a timer, only two hourglasses. One measures four minutes and the other measures seven minutes.

Problem: How can you cook the egg for exactly 9 minutes using only these two hourglasses?

2. You have two pieces of rope. You know that each one takes exactly one hour to burn. They are not necessarily of the same length or width as each other. They also are not of uniform width (for example, they may be wider in the middle than at the ends). Thus, burning half of the length of the rope does not necessarily take half an hour.

Problem: By burning the ropes, how can you measure exactly 45 minutes worth of time?